LIBRARY, NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

LOCAL AREA NETWORK TERMINAL MANAGEMENT
IN SUPPORT OF STOCK POINT LOGISTICS
INTEGRATED COMMUNICATIONS ENVIRONMENT (SPLICE)

by

Jerry D. Barnes
December 1982

Thesis Advisor: Norman Schneidewind

Approved for Public Release; Distribution Unlimited

T207791

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Local Area Network Terminal Management in Support of Stock Point Logistics Integrated Communications Environment (SPLICE)	5. TYPE OF REPORT & PERIOD COVERED: Master's Thesis, December, 1982 / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Jerry D. Barnes	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93940	12. REPORT DATE: December, 1982 / 13. NUMBER OF PAGES: 40
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Terminal Management, Virtual Terminal Protocol, Network Virtual Terminal, SPLICE, Local Area Network

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis examines the questions of user requirements, design considerations, and network environment for a local area network Terminal Management function in support of the Naval Supply Systems Command's Stock Point Logistics Integrated Communications Environment (SPLICE). Criteria are developed from this examination. They include process-process communication, virtual terminal, and user defined screen capabilities as well as a negotiated (Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT (Continued) Block # 20

virtual terminal protocol based upon a network virtual terminal concept. Recommended generic and specific models of the Terminal Management function applying these criteria are then presented.

Approved for public release; distribution unlimited.

Local Area Network Terminal Management in Support of Stock Point Logistics Integrated Communications Environment (SPLICE)

by

Jerry D. Barnes
Lieutenant Commander, Supply Corps, United States Navy
B.S., Oregon State University, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
December 1982

LIBRARY, NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93940

ABSTRACT

This thesis examines the questions of user requirements, design considerations, and network environment for a local area network Terminal Management function in support of the Naval Supply Systems Command's Stock Point Logistics Integrated Communications Environment (SPLICE). Criteria are developed from this examination. They include process-process communication, virtual terminal, and user defined screen capabilities as well as a negotiated virtual terminal protocol based upon a network virtual terminal concept. Recommended generic and specific models of the Terminal Management function applying these criteria are then presented.

TABLE OF CONTENTS

I. INTRODUCTION 8
A. OBJECTIVES OF RESEARCH 8
B. BACKGROUND 9
C. ADDENDUM TO REINHART AND ARANA THESIS 10

II. REQUIREMENTS DEFINITION 13
A. OVERVIEW 13
B. STOCK POINT REQUIREMENTS 14
C. NAVSUP POLICY AND DIRECTION 16
D. CONCLUSIONS AND ASSUMPTIONS 17

III. ASPECTS OF TERMINAL MANAGEMENT 19
A. OVERVIEW 19
B. TM TECHNICAL CONCERNS 19
C. LOCAL AREA NETWORK (LAN) ENVIRONMENT 21
D. GENERIC TM MODEL 22
E. TERMINAL MANAGEMENT APPROACHES 24
1. Parametric Approaches 24
2. Virtual Terminal Protocols 26
F. REINHART AND ARANA TM 29

IV. RECOMMENDATIONS 31
A. OVERVIEW 31
B. GENERAL RECOMMENDATIONS 31
C. RECOMMENDED TM MODEL. 32
1. Input/Output Device 32
2. Local Virtual Terminal (LVT) 33
3. Network Virtual Terminal (NVT) 33
4. Virtual Terminal Protocol (VTP) 34
5. Session Example 34

V. THE RECOMMENDED APPROACH 36
A. THE ASSUMPTIONS 36

B. THE CONCEPT 36
C. TM IMPLICATIONS 37

LIST OF REFERENCES 38

INITIAL DISTRIBUTION LIST 40

LIST OF FIGURES

1.1 Possible Logical LAN Connections 11
3.1 Possible Physical LAN Connections 21
3.2 Generic Terminal Management Model 22
4.1 Recommended TM Model 32

I. INTRODUCTION

A. OBJECTIVES OF RESEARCH

The objective of the Stock Point Logistics Integrated Communications Environment (SPLICE) Local Area Network (LAN) Project at the Naval Postgraduate School, Monterey, was to develop alternatives for SPLICE LANs. The thesis submitted by Lieutenant Joesph N. Reinhart III, USMC and Lieutenant Ricardo Arana, Peruvian Navy [Ref. 1], concerned itself with the development of functional design specifications for the implementation of the Database and Terminal Management functions of a functionally distributed LAN in response to the requirements of the Naval Supply System's stock points and inventory control points.

The objectives of this thesis research is to further define the requirements of the SPLICE users and to develop from these needs a generic model of Terminal Management (TM) functions necessary to support the services required. The rationale for using a generic model rather than a specific model lies in the evolutionary state of Naval Supply Systems Command (NAVSUP) data processing objectives. In an executive level briefing, the SPLICE project office stated :

> We cannot afford the luxury of supporting "Navy unique" software packages (in the future). We simply cannot afford the resource draw (drain). Our policy must not allow unique solutions. Our systems must fit within the technology and capabilities of the general ADP marketplace. [Ref. 2]

In keeping with that policy, this thesis will attempt to recommend, using the Reinhart and Arana thesis as a theoretical foundation, a TM functional specification capable of supporting presently envisioned SPLICE LAN configurations,

while presenting the capability to support evolutionary configurations of the future.

B. BACKGROUND

SPLICE, as concieved by NAVSUP describes a near-term system to provide badly needed local and system network communication and management functions without further overloading the present host system. It also presents the conceptual foundation for respones to future changes to both customer requirements and technological advances. SPLICE draws together under one conceptual umbrella the myriad of new applications being developed independently throughout the supply shore establishment.

SPLICE, in its simplest form, is designed to provide a hardware and software architecture capable of supporting a wide variety of interactive application programs on both local and remote terminals. When fully realized this capability will significantly reduce the proliferation of stand alone computers at support sites presently unable to obtain data processing services otherwise.

Significant factors which will determine the success of the SPLICE concept will be the speed and accuracy of data and file transfers within and between LANs and the speed, accuracy and ease of interactive terminal sessions. It is the latter concern which is the reason for this thesis.

NAVSUP and Fleet Material Support Office SPLICE documentation provide detailed information on SPLICE software design considerations [Ref. 3], systems specifications [Ref. 4], functional design [Ref. 5], and telecommunciations plans [Ref. 6]. References 7 and 8 provide insight into the magnitude and variety of transactions contained in typicial applications which SPLICE will be expected to support. As the LAN design project, with which this thesis is

associated, is not constrained by SPLICE designs and specifications, the above references were used primarily as a source of purpose and objectives. Reference 1 provides a very readable synopsis of References 3 through 6 . Readers desiring such information are referred to sections IA and IC of that document. In keeping with the objective of their research, Reinhart and Arana presented their recommendations for Database and Terminal Management functional specifications in support of a LAN. In the name of brevity, a lengthy condensation of that thesis will not be provided in this thesis. Instead, as this thesis uses the Reinhart and Arana thesis as a jumping-off point, later developments and/or information that might modify that thesis will be presented in section C of this introduction.

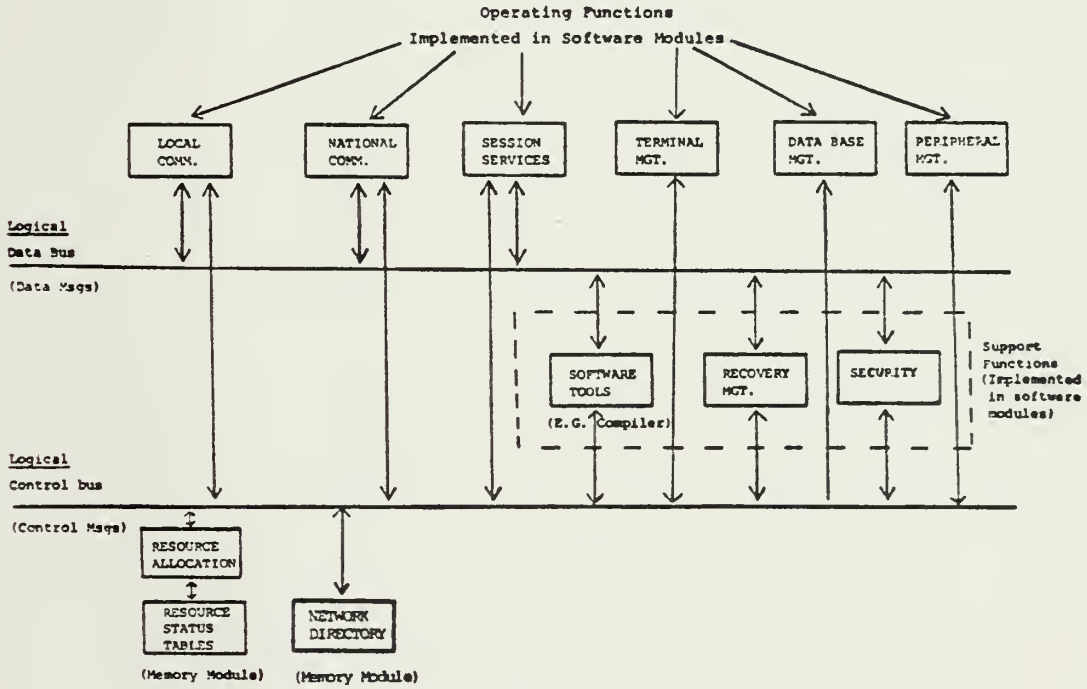
Additional insights into the variety and size of the tasks expected to be processed on the SPLICE LAN were gained by personal contacts with persons attached to the SPLICE project office at NAVSUP headquarters in Washington, D.C. and functional managers at Naval Supply Centers in Oakland and San Diego. The intent of this thesis, and therefore these interviews, was to establish in the author's mind a generic definition of interactive processing requirements so that Terminal Management (TM) alternatives might be evaluated. The results of these investigative interviews are contained in Chapter II of this thesis.

C. ADDENDUM TO REINHART AND ARANA THESIS

In their background section, Reinhart and Arana implied that a commerical product named Terminal Application Processing System (TAPS) was the most probable implementation of terminal management, database management, complex management, and transport management functions. At the time, that was a valid assumption and shared by NAVSUP.

Actual responses have shown that a significant number of vendors responding the the SPLICE solicitation prefer their own functional modules to TAPS and are bidding accordingly. This action would appear to support Reinhart and Arana's contention that a functionally distributed LAN without a central complex manager is viable.

Section ID of the thesis discusses SPLICE functional implementation and provides a diagram in its Figure 1.3 of a possible implementation. An updated version of that diagram is contained in Reference 9 and Figure 1.1 below.



[Ref. 9: p.4-6]

Figure 1.1 Possible Logical LAN Connections.

Also discussed in section ID of Reference 1 are the relative merits of a high level database query language versus the interactive application programs approach of NAVSUP. The thesis implies that NAVSUP should, and therefore has yet to, investigate the feasibility of a database query language capability within SPLICE. This author's interviews and a review of NAVSUP policy guidance indicate that NAVSUP has the eventual development and use of such a query language high on their ADP priority list.

II. REQUIREMENTS DEFINITION

A. OVERVIEW

The method employed in attempting to arrive at a requirements definition for the Terminal Management (TM) function was a series of interviews using standard questions. The interviews were conducted with persons at the Naval Supply Centers at Oakland and San Diego. These centers were chosen primarily because of locale, but this choice does not diminish the effectiveness of the interviews. Between them, San Diego and Oakland, conduct the entire range of stock point operations, with the exception of strategic forces support. San Diego supports a major fleet presence, a large training command, two major air stations, and numerous shore maintenance and administrative commands. Oakland supports a smaller fleet presence, one air station, fewer shore establishments, but serves as a clearing house for Western Pacific requirements. Both centers presently support remote terminal operations. Both have implemented either or both IDA or/and APADE to varying degrees. These terminal based interactive application programs are currently executed on stand-alone minicomputers, but their mere existence allows the functional managers using them to answer questions not answerable by managers without experience with this type of approach.

The impressions garnered from these interviews are contained in section B below. Section C discusses information gained in meetings with NAVSUP SPLICE project office personnel. Section D summarizes the user requirements and associated assumptions that will be used throughout the remainder of this thesis in evaluating TM approaches.

B. STOCK POINT REQUIREMENTS

Interviews with stock point functional managers quickly established the presence of managerial difficulties expected in trying to hold together an ongoing production effort based on established procedures and technology while trying to simultaneously implement a newer technology. Despite their tribulations, these managers were most willing to share their experiences to date with terminal based interactive application programs.

Both IDA and APADE are designed to utilize a menu-driven form mode of interactive data entry and file inquiry/update. Both have extensive process options available to the terminal user.

The APADE users, primarily in the purchasing division of the Procurement Department, logically and physically separate data entry functions from data inquiry/update functions. This is based mostly upon time constraints of data entry and the volume of these transactions. The TM implications of this separation is that the data entry clerks would be best served by the least creative, least complicated TM that would still support interactive form data entry, i.e., when finished with one form, the user wants another form immediately, not a helpful but nevertheless key-stroke consuming menu. On the other hand their co-workers who are responsible for handling a rather large number of document inquiries from not always patient customers would find the ability to split their screen into multiple sections and to conduct a separate inquiry or update in each section very helpful. The purchasing administrative personnel are frequently asked to produce information (written and oral) that requires multiple access to files and/or manual manipulation of the data accessed. The primary cause of their effort is that they are

constrained by the formats, both display and reports, designed into the system. The capability to design a multiple-use screen and report format at the terminal is indicated.

The IDA users, financial accounting, commonly referred to as Triple-A, do not separate their data entry from inquiry/update. Each clerk who has a terminal has often found himself in the situation of needing to view a record from more then one file. This situation is especially prevalent when reconciling vendor invoices with requisitions. This need is presently met by calling a clerk who has access to the necessary files (APADE or UADPS) and passing the necessary information by phone -- automation indeed!

The Customer Services personnel primarily interface with the UADPS applications residing on the Burroughs host mainframe. Their interaction with the computer is varied, but basically inquiry in nature. Data entry is normally batch processed, and will in all likelihood remain so until OCR technology replaces the current card readers. Because most queries are made to records stored and processed in 80 column card format, a scroll-mode terminal presentation with the ability to continually enter queries and direct replies to a nearby printer is the near concensus choice for terminal interaction. A less frequent activity conducted by Customer Services personnel requires them to spend hours researching and cross referencing records from various UADPS and financial files (now IDA files). This reasearch is normally conducted by the most experienced and senior clerks and supervisors in the division, so it would seem that countless hours could be saved, not to mention dollars, if these persons could break away from the standard displays to which they are now limited and set up a screen display suitable for researching and controlling the source of display input.

In warehousing and material receipt functions, the use of automation has just begun, primarily in stowage and retrieval operations. The day of the use of bar codes and light pens for receipt processing and inventory management is still somewhat distant. Although there is good reason to suspect that these devices should be considered peripherals and therefore not within the purview of this thesis, the author can envision a scenario where they would be direct input devices to several UADPS application programs. As such, the author has chosen to include them in the category of potential terminals.

C. NAVSUP POLICY AND DIRECTION

In addition to a survey of stock point personnel, a trip was made to Washington, D.C. to meet with SPLICE project office personnel. The objective of the meetings was to gain an appreciation for the direction of SPLICE LANs. Although the development of a generic TM model from requirement needs is deemed academically justifiable, there remained the concern that to stray too far from the realities of the Naval Supply System would result in an academic exercise of little import. The quotation cited in Chapter I, Objectives of Research section, convinced this author that such a model could in fact be of use, particularly in view of the desire to move away from "Navy unique" software. The functional TM model presented in Chapter IV represents the author's desire to produce a recommendation for the near future, and therefore embraces the reality of the SPLICE environment more than the generic TM model presented in Chapter III.

Additional information gained from these meetings included familiarization with the long-term objectives of SPLICE implementation. Although these objectives do not fit the category of user requirements, they are paraphased below

because failure to keep them in mind in a design process increases the probability that the resultant design may too restrictive to be of value in meeting future growth.

NAVSUP SPLICE Project Objectives

- Data must be made available to the customer, easily accessed from wherever the customer is located on a continuous basis around the clock.
- A distributed data base must be developed.

Goals for NAVSUP ADP

- All punched cards must be eliminated.
- The use of point-of-sale terminals and hand held terminals with optical scanning capabilities at data entry points.
- The use of bar codes and magnetic strips in inventory control and warehousing applications.

D. CONCLUSIONS AND ASSUMPTIONS

The following conclusions regarding the TM requirements of users of the SPLICE LAN are based upon the interviews discussed above and observations of terminal use at stock points. Assumptions are based upon the same observations coupled with the author's experience in stock point operations.

1. A TM must be able to support a wide variety of terminals, varied not only in make/model, but also in operational characteristics.

2. A TM must be able to support character, line, page, scroll, and form mode editing capabilites at the same terminal.

3. A TM should provide the ability for a user to locally format a display area, and resultant hardcopy output.

4. A user should be able to simultaneously display multiple processes utilizing different application programs.

5. A TM should provide the mechanism to allow application to application interaction (author's assumption)

6. A TM should be able to support non-interactive data entry, such as light pens and hand-held OCR (author's assumption).

7. A TM should provide a general framework within which future terminal functions can be accomodated (author's assumption).

8. A TM design must recognize the relative lack of sophistication of potential users and the probability of high turnover in these positions.

III. ASPECTS OF TERMINAL MANAGEMENT

A. OVERVIEW

In addition to the user requirements discussed in Chapter II, investigation and evaluation of alternative TM techniques requires an appreciation for the technical concerns of a TM. Section B discusses these concerns. Section C outlines the LAN environment assumed for this thesis. Section D presents the author's view of a generic TM model. Section E then explores some of the TM approaches cited in contemporary literature. Section F contains a review of the TM approach advocated by Reinhart and Arana [Ref. 1].

B. TM TECHNICAL CONCERNS

TM, as a list of widely agreed to discrete functions and protocols, does not exist. Rather, there exists an impressive, if somewhat confusing, spectrum of alternative methods of implementing a TM. On the most ambitious end of this spectrum is a TM module which provides the full range of functions described in the Presentation Layer (layer 6) of the International Standards Organization (ISO) Open Systems Interconnection (OSI) reference model. One of the many definitions of the task of the Presentation Layer "...is to support communications by providing commonly known virtual devices and commonly known virtual information to the distributed applications" [Ref. 10: p. 227]. A less ambitious TM would be expected only to "hide terminal idiosyncracies from the application programs" [Ref. 11: p. 484]. The latter representing a subset of the former.

It is helpful to try to describe the design concerns of a TM and then to blend these concerns with user requirements to form a model of the ideal or generic TM which would recognize both the designer's and user's concerns.

A suggested list of concerns of the TM designer is found in Reference 12 (pp. 82-84). These concerns are:

1) Control of the Terminal Handling - Assuming the wide variety of terminals and attendant variety of characteristics, the parameters which affect local handling of a terminal must be known to the TM, and to no other LAN module nor to remote users.

2) Dialogue Mode - The TM should provide methods for selecting/providing support for both half-duplex or full-duplex operations.

3) Terminal Data Structure - Like terminal control characteristics, the TM must be aware of the data structure parameters of the terminal as well as the command language primitives available for manipulation of that structure.

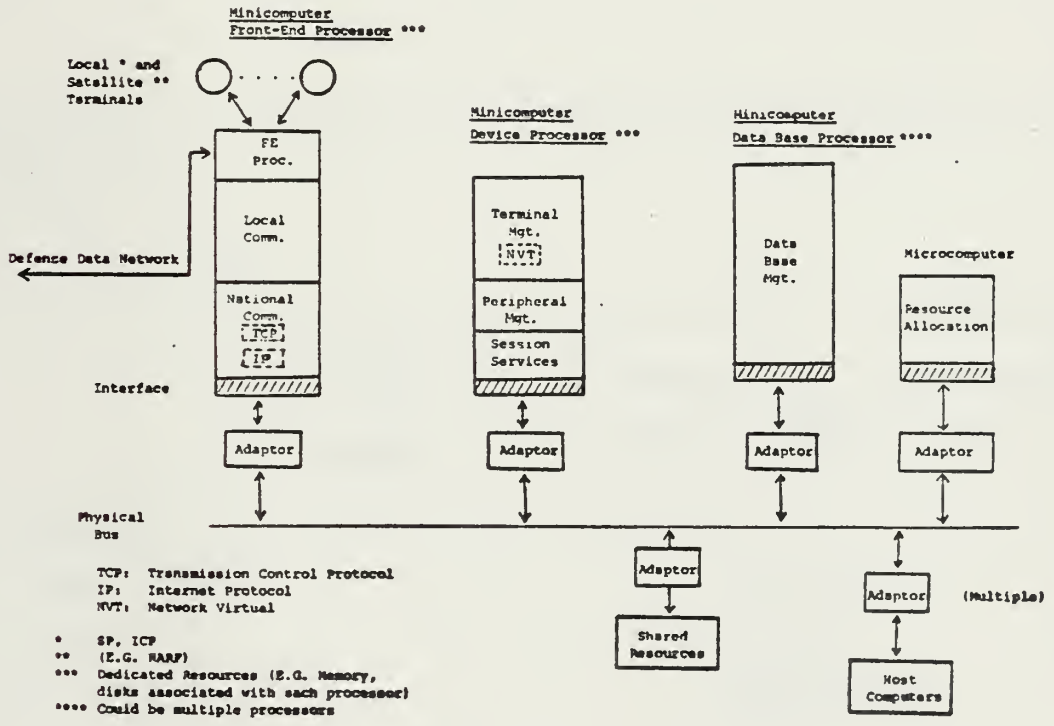
4) Symmetry - The design of the TM must be concerned with the desirability of symmetrical forms of connection, e.g. process-process and terminal-terminal interactions

5) Negotiations - The TM must provide a dynamic mechanism to negotiate the facilities and parameters to be used in each interconnection.

6) Attentions - The TM must be capable of interpreting and handling the variety of methods used in terminals and processes to signal "break", "abort", and other such attention commands. These signals are normally expedited signals "out-of-band" or outside the normal flow of data.

C. LOCAL AREA NETWORK (LAN) ENVIRONMENT

The LAN environment in which the TM being discussed in this thesis will operate is a fully distributed LAN based upon seven primary functional software modules; local communications, national communications, front-end processing, session services, terminal management, database management, and peripheral management. Figure 1.1 introduced the logical connections of this LAN. Figure 3.1 presents a possible physical connection configuration.



[Ref. 9: p. 4-7]

Figure 3.1 Possible Physical LAN Connections.

This thesis also assumes a multiplexed data/control bus capable of supporting half and full-duplex communications. Further, it is assumed that the terminals connected or potentially connectable to this LAN are hetrogeneous and that the hetrogenous mix will be in a constant state of flux during the next decade.

D. GENERIC TM MODEL

Given the user requirements, design concerns, and environmental assumptions developed above, Figure 3.2 presents the author's attempt to meet these criteria with a generic TM model.

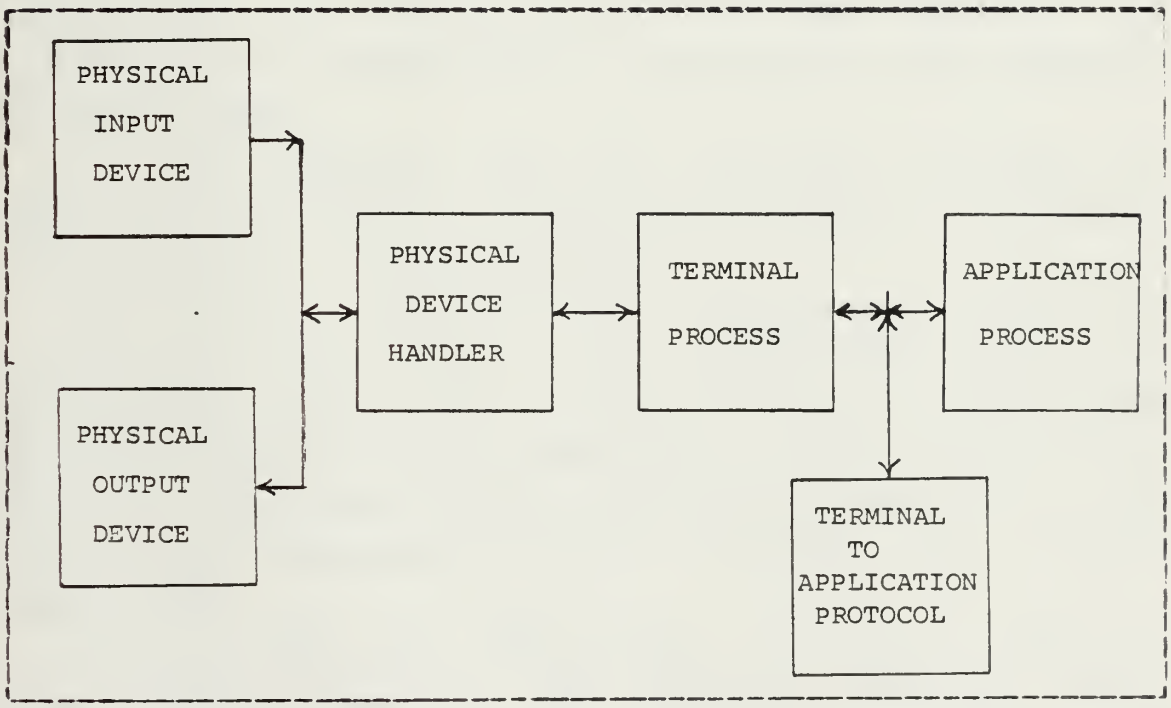


Figure 3.2 Generic Terminal Management Model.

Figure 3.2 has the following components:

1) A generic terminal with an input capability (keyboard, light pen, optical scanner, etc.) and an optional output device (CRT screen, signal light, teletype, etc.).

2) A device or module which can understand the signals coming from the input device and can send signals to the output device which it can recognize. This component also has a command language of its own which allows the user to design an output display and to map more than one process to that display.

3) This component represents the terminal in dealings with the application process (one per process). It also builds and manipulates a terminal data structure for each process.

4) This component represents the application program in dealings with the terminal process (one per process). It also builds and manipulates an application data structure for each process.

5) This component sets the rules for and format of communications between the terminal processes and application processes (for all processes).

In the most simplistic terms components 3 and 4 are attorneys for their clients, the terminal and the application program, respectively. They, and only they, know their clients capabilities and expectations. Both are very strong willed and therefore need an arbitrator (component 5) to ensure that communicatins are meaningful and properly coordinated.

At this point this model simply provides an ideal TM whose generic components, if implemented ideally, would provide a modular TM capable of meeting user requirements and design concerns while having the flexibility to respond to changes. A recommended specific implementation of this model will be presented in Chapter IV.

E. TERMINAL MANAGEMENT APPROACHES

Approaches to TM can be generally discussed in two broad, and unfortunately not mutually exclusive, categories; parametric and virtual terminal.

1. Parametric Approaches

Generally, parametric terminal protocols attempt to list a set of terminal characteristics with each type of terminal having a different set of parameters for each characteristic. The host computer may then set the parameters available on that terminal to values needed for the process/application [Ref. 12: pp. 84-86].

This approach is used by the ARPANET Telenet protocol and by systems implementing International Consultative Committee for Telephones and Telegraphs (CCITT) recommendations.

When evaluating these two approaches the ARPANET Terminal Interface Processor (TIP) is compared to the CCITT Packet Assembler/Disassembler (PAD). Both approaches use a primitive command language to open and close connections and to set terminal parameters. The primary difference being the perceived role each plays in the network architecture. In the ARPANET the TIP is a limited capability logical host with knowledge of terminal parameters. Whereas, the CCITT considers a PAD an integral part of the network acting as an interface between data terminating equipment (DTE). A DTE can be either a host or a terminal [Ref. 12: pp. 84-86], [Ref. 13: pp. 335-348], [Ref. 14: pp. 586-587].

The PAD cited in CCITT protocols is the most comprehensively defined parametric approach to terminal handling. There are three basic approved recommendations covering operation of the PAD; X.3 which defines the PAD itself, X.28 defines the interface between the terminal and the PAD, and

X.29 defines the interface between the PAD and the host [Ref. 12: pp. 84-86].

The shortcoming of these approaches is that the PAD protocols provide no generic functions. They assume that the application in the host knows what the terminal will do and that the terminal will do what is intended [Ref. 14: pp. 586-587].

Telenet's Interactive Terminal Interface (ITI) provides an enhanced set of parameters which helps offload some of the terminal handling responsibilities from the host. Telenet offers a further refinement called a virtual terminal which includes a few generic functions on top of the ITI parameters. Unfortunately, the ARPANET virtual terminal was designed primarily to support scroll-type terminals which have much fewer parameters than more sophisticated page and form mode terminals. Although the list of parameters was extended, few of the options (parametric values) were implemented [Ref. 14: pp. 586-587].

Both these approaches are most suitable for providing TM for existing terminals, the more homogenity the better. To be really useful, these functions should be standardized so that the host system can rely upon a PAD/interface with known properties. Even with standarization the envolvement of the host system in TM would still be extensive or the list of PAD/interface parameters would be enormous [Ref. 13: pp. 347-348].

As several of the user requirements and design concerns discussed earlier imply the need for the greatest amount of transparency achievable and further imply an increasing number and variety of terminals, the parametric approaches appear too limited.

2. Virtual Terminal Protocols

Virtual Terminal Protocols (VTP) have undergone significant evolution since the first VTP was placed in use by the ARPANET. This VTP was designed primarily with scroll-mode terminals in mind. It is based upon three basic principles; the concept of a 'network virtual terminal', the concept of negotiation of options, and a symmetrical view of terminals and processes [Ref. 12: p. 88] [Ref. 14: p. 588]. This first VTP laid very firm ground for further sophistication of the Virtual Terminal (VT). Unfortunately, although the Telenet VTP was designed with fifty-eight parameters, very few were actually implemented. It therefore remains to explore a few more of the many VTP approaches developed since the ARPANET VTP.

A model which focuses on page and form mode terminals was developed by Schicker and Duenki. It is used in the European Informatics Network (EIN) and is described in Reference 15 (p. 485). This model is called a data structure model. In it a data structure is viewed as containing a set of fields each of which has certain attributes such as size of the field, what type of characters it contains, whether or not the field can be modified by the user, etc. This definition of a data structure has become widely accepted and will be used in the remainder of this thesis. This model assumes that application programs are written to perform abstract operations on a data structure and that the remote (user) process has a similar data structure. The VTP is the mechanism by which the changes made by the application process to its data structure are passed to the user process so that its data structure can be changed accordingly, and vice versa.

A refinement of the data structure model is described in Reference 13 (pp. 362-365). In this model a terminal has a data structure and a controlling process called a "T-PAD" with a relationship much like that described in the parametric approach in subsection 1 of this section. Similarly, the host system has a data structure with which it is designed to interact via a controlling process called an "S-PAD". The messages passed between these two "PADs" to negotiate the data structure and the available commands to manipulate the data structure are contained in the VTP. The appeal of this approach is two-fold; the S-PAD implies a NVT concept such as described by Tanenbaum in Reference 15 (p. 423), where a NVT is an abstract terminal the characteristics of which are assumed by all interactive application programs; secondly, a symmetrical approach such as this allows not only the traditional terminal to application interaction, but also terminal to terminal and application to application interaction, given a sufficiently adept VTP. The utility of such capabilities can be shown in a common Stock Point scenario. This scenario exists when an application program, such as APADE, creates records that are duplicated or entered into files necessary to the correct operation of another application program, such as IDA. For instance, the establishment of a contract, which requires a record update in an APADE file, also necessitates an update of a financial record in an IDA file. Such changes can be made by batch processing (current practice) or by application to application interaction such as made possible by a TM implementing this type of model.

The approach above is very similar in function, if not vocabulary, to the NVT envisioned for ARPANET's Telenet protocol and the INWG VTP. All are deemed symmetrical in that each side of a session has its own view of the state of the VT [Ref. 14: p. 589]. This as opposed to an

asymmetrical model where the VT is considered only from the perspective of the application program. In such a model the physical terminal is transformed by software to appear as a VT to the application program [Ref. 16: p. 304]. This approach cannot support terminal to terminal or process to process interaction [Ref. 14: p. 588].

In each of the VT concepts described the VT's are supported by two elementary protocols; a virtual terminal display data transformation protocol and a control protocol. The data transformation protocol maps display commands from the sending process into the prescribed input data formats for the receiving process. The control process exchanges non-display information for coordinating interactions [Ref. 17: p. 85].

A more expansive approach is offered in Reference 10. In this approach three abstractions (virtualizations) are proposed; a virtual device, a conceptual data type defintion, and a conceptual image. The virtual device is considered an association between a definition of a structure for a (device) data object and a set of operations which are the only means for accessing this (device) data object. The conceptual data type definition is a similar association, but with regard to the structure of data and the operations which may be peformed on data structure objects. The conceptual image is considered a definition of the means by which a mapping of the conceptual data on the virtual device is obtained. The thrust of this concept is that in a hetrogenous network, assumptions regarding indentical virtual devices and data structures may not be desireable and possibly not practically viable. Only an agreement of negotiated parameters need be known by each partner. The authors wrote a follow-on article, [Ref. 18], in which they presented a detailed recommendation of the protocols and options for each of these virtualizations.

All of the above approaches require the negotiations of options/parameters to be used in a session, be they device characteristics, data structure or commands. Negotiations are either asynchronous or master-slave (synchronous) depending on the symmetry of the interaction and the trust the designers place in whatever mechanism they may have implemented to resolve negotiation deadlocks. The literature regarding negotiation algorithms seems polarized with each side singing the praises of their approach. The author's preference is included in the TM model presented in Chapter IV.

F. REINHART AND ARANA TM

In Reference 1 (p. 55), the authors proposed a TM approach based upon a "Virtual Terminal" management concept. The primary feature of their VT is that it "... converts a single physical terminal into multiple virtual terminals, each of which may be written into or queried for input". To support this concept the idea of a user defined screen configuration is proposed. This would allow a user to divide his screen into "windows" each of which contains the display of a separate process. Although only one window/process would be active at a given moment, it is clear that the implementation of this concept would satisfy several of the requirements and concerns addressed earlier in this thesis.

The thesis also discuss on page 74 the use of a generic terminal transformation table. This table, as proposed by Hillsberg [Ref. 19] is used to convert specific physical terminal sub-functions to generic commands and vice versa.

Although this thesis owes its roots to the Reinhart and Arana effort, subsequent readings and investigation have led this author to step back from most of the specifics

presented in their thesis and move toward a broader concept for TM functional specification using their concepts as an inspiration.

IV. RECOMMENDATIONS

A. OVERVIEW

The recommendations contained in this chapter are a marriage of the generic TM model presented and the variety of specific approaches discussed in Chapter III. The result is a specific model consisting of components and protocols interfacing those components. General recommendations are listed in section B below. Section C presents the recommended model.

B. GENERAL RECOMMENDATIONS

1) The TM should be based upon the concept of a NVT. Each application program should be able to assume that it is dealing with a terminal where the default parameters will be used unless the user negotiates different parameters using the NVT capability.

2) The TM should support the highest level of abstraction technologically available.

3) Negotiation of device definitions should be hierarchical. Standard classes of terminals defining mandatory characteristics (minimum parameter values) should reside at the highest levels of the hierarchy. Optional (negotiatible) operations should reside at lower levels.

4) The TM should support user defined screen format for use in both displaying multiple processes and designing display and possible report format.

C. RECOMMENDED TM MODEL.

Figure 4.1 presents the Terminal Management Model. recommended by the author. The components are explained below.

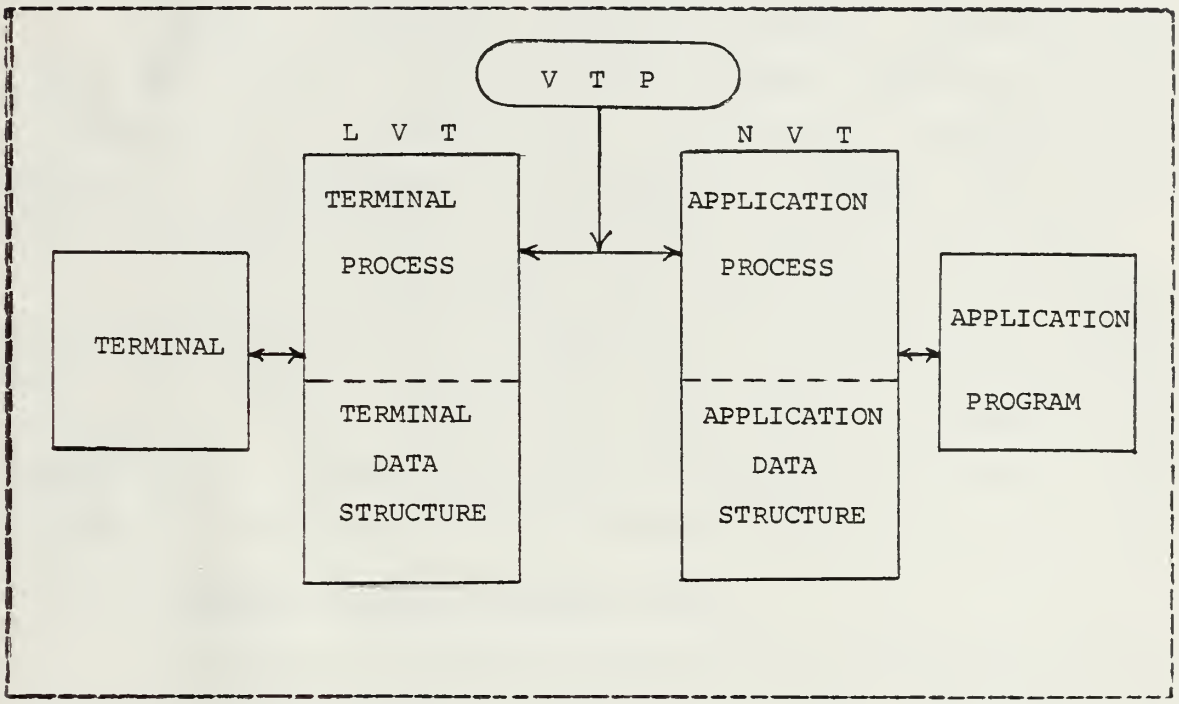


Figure 4.1 Recommended TM Model.

1. Input/Output Device

A terminal is viewed as two separate devices. The separation recognizes that a LAN may recieve input from devices that have no output capability, e.g. light pens, OCR scanners, etc. The TM must be able to handle inputs from such devices. Additionally, the ability to transform and control inputs is conceived to be totally separated from the ability to control and map output.

2. Local Virtual Terminal (LVT)

The function of the LVT is to hide the idiosychracies of the user terminal from the LAN and to map its data structure and any changes to that structure to the terminal display device. The LVT consists of a terminal process (TP) component and a disk-based terminal data structure (TDS).

The TP utilizes a generic terminal transformation table (see Ref 19) to virtualize terminal sub-functions and a hierarchically organized set of terminal primitives and parameters as discussed in paragraph B.3 above. The TP also uses a user defined map of the screen display for output.

The TP is responsible for representing the terminal in negotiations with the application process (or another TP). It is also responsible for mapping terminal input commands to the TDS and for mapping TDS changes to the output unit. These functions are comparable to the "T-PAD" discussed in Reference 13 and Chapter III.

3. Network Virtual Terminal (NVT)

The NVT is configured identically to the LVT with an application process (AP) in place of the TP. The AP has the same data structure modifications responsibilities, except that it takes orders from and reports changes to an application program as opposed to a terminal.

The major conceptual difference between the NVT and the LVT is that the NVT is a parametric model of the entire network's concept of a terminal. Its primitives and parameters are fixed. The application program's requirements are met by passing necessary parameter values to the AP. Note that the number of parameters are fixed, but that each parameter may offer more than one option (value). The AP then represents the application program in negotiations with the TP as to what values will actually be used. Obviously,

the AP will not agree to any parametric values lower than the program requires, but it may agree to higher values should the TP insist. The AP hides all negotiatied values from the program except those that the program expects.

4. Virtual Terminal Protocol (VTP)

The VTP is a message based protocol with several functions, discussed here in chronological order. The VTP controls the the exchange of negotiation messages between the TP and AP. During these negotiations the primitives and parameter values of control messages are selected as well as the contents of the data structure. The recommended negotiation protocol will be explained in the session example below. Once all values are set, the VTP becomes responsible for passing control messages between the TP and AP for the manipulation of their respective data structures. In fulfilling this responsibility, the VTP controls the sequencing of messages according to the communication method selected during negotiations, i.e., alternating or free-running, and dictates the format of these messages.

5. Session Example

a) The user logs on his terminal identifying himself and the terminal ID. Terminal ID can be done automatically if the capability exists.

b) The TP will compute the command signals needed to handle this particular terminal using the transformation table. The TP will then begin a screen formatting subroutine conversation with the user, in which the first user response may result in a default screen. (note: asking the question rather than waiting for the user to request screen formatting should evoke curiosity and quicken the learning process for this feature) This routine will construct the output map for the TP's subsequent interaction with the output device.

c) Once the user has defined the screen format desired, the first (and possibly only) command to call an application program is entered.

d) The TP passes this message to the LAN which may need to send it out to the long-haul network.

e) The destination node will establish an AP to which the application program passes its terminal and data structure requirements. At this point, and throughout the negotiation process, the VTP is ensuring one-way communications.

f) Once negotiations have been completed, the application program directs the AP to set the initial state of its data structure. Upon doing so, the AP, using the VTP message format, passes the agreed data structure instructions to the TP which both creates an identical initial state in its data structure and maps the data structure to the user-defined screen format.

g) At this point the VTP may allow free-running (asychronous) communications if both parties can support such.

h) When the application program is completed, the connection is severed and the TP begins its user dialouge anew.

Admittedly, this TM model is a compromise between the generic goals presented in Chapter III and the practical implications of NAVSUP's dedication to application programs. A more esthetically pleasing concept is outlined in Chapter V.

V. THE RECOMMENDED APPROACH

A. THE ASSUMPTIONS

The concept discussed in this chapter is a recommendation for the future. It is based on the following assumptions:

1) NAVSUP is willing to abandon the development and use of application programs in favor of functional modules and a distributed database system.

2) NAVSUP defines the set of queries and reports it wishes the system to provide.

3) NAVSUP is willing to allow increased local flexibility in the design of display and report formats.

B. THE CONCEPT

Given the above assumptions, the envisioned data base system would be built by use of a data base "designer" and program generator such as discussed in Reference 20 . The basic tool of this building process is called a Hierarchical Interactive Query (HI-IQ) language. The result of this process would be a database capable of supporting programs written in a COBOL Data Manipulation Language (DML) for data entry and inquiry. The primary charm of this concept is the reduction of redundant record fields in different application files and the concomitant reduction in required file space. Given that there are only two primary fields forming the backbone of the vast majority of stock point and inventory control point transactions, the requisition number (order number) and the National Stock Number (or part number), the concept of a hierarchical data base system to support what is now supported by UADPS file management programs is not so far-fetched.

A second enhancement this approach offers is the elimination of the need for application to application interaction, be it batch or interactive. Because the financial data field for a UADPS requisition entry is the same field used in IDA, the need for passing this information to the IDA application program for entry is eliminated.

C. TM IMPLICATIONS

Adoption of the system described above would not necessitate scraping the TM recommended in Chapter IV, but to fully capitalize on the benefits of the system, certain changes would be necessary.

The most important change would be to enhance the Terminal Process (TP) by providing it with a data structure description language which would logically be a subset of the system's Data Manipulation Language (DML). This language would be available to a user to name fields and zones and for defining the attributes and dimensions of such. This capability would enable users to easily design screen and report formats tailored to their needs. But, such a capability would require a rethinking of the application program's master role in establishing data structure parameters.

As this concept would move away from the application emphasis on form or page designs in data structures, changes would also be necessary in NVT and VTP parameters. Both could be simplified because they would not have to be structured to support a myriad of often conflicting application programs.

Should the assumptions in section A evolve , the concept outlined in this chapter is strongly recommended for further study.

LIST OF REFERENCES

1. Reinhart, Joesph N. III and Arana, Ricardo Database and Terminal Management Functional Design Specifications in Support of Stock Point Logistics Integrated Communications Environment (SPLICE), Master's Thesis, Naval Postgraduate School, June 1982.

2. Informal communications with Naval Supply Systems Command personnel, September, 1982.

3. U.S. Navy Fleet Material Support Office, Environment Division: Code 9441, Stock Point Logistics Integrated Communications Environment (SPLICE), Software Design, 19 March 1979.

4. Fleet Material Support Office, Department of the Navy, Document No. F94L0-001-9260-SS-SU01, Stock Point Logistics Integrated Communications Environment (SPLICE) System Specification, 2 February 1981.

5. Fleet Material Support Office, Department of the Navy, Document No. F94L0-001-9260-FD-SU01, Stock Point Logistics Integrated Communications Environment (SPLICE) Functional Description, 1 May 1980.

6. Naval Supply Systems Command (Code 0415), Logistics Telecom- munications Branch, SPLICE Telecommunications Subsystem Project Plan (TSPP), 9 March 1981.

7. Naval Supply Systems Command (Code 041), Integrated Disbursement and Accounting Data Exchange Systems Capability Study - Interim Report, March 1981.

8. Naval Supply Systems Command (Code 044), APADE II Systems Capability Study, May 1981.

9. Schneidewind, Norman F. Functional Design of A Local Area Network for the Stock Point Logistics Integrated Communications Environment, Naval Postgraduate School Document No. NPS 54-82-003, December 1982.

10. Schindler, Sigram; Flasche, Ute; and Bormann, Carsten "The Presentation Service Model", Computer Communications, Vol.4, No.5, October 1981

11. Tanenbaum, Andrew S. "Network Protocols", ACM Computing Surveys, Vol.13, No.4, pp. 453-489 ,December 1981.

12. Day, John D. "Terminal, File Transfer, and Remote Job Protocols for Hetrogeneous Computer Networks", Protocols for Data Communication Networks, Franklin F. Kuo, Editor pp. 78-118, Prentice-Hall, Inc., 1981.

13. Davies, D. W.; Barber, D. L.; Price, W. L.; and Solomonides Computer Networks and Their Protocols, John Wiley and Sons, Inc., 1979.

14. Day, John D. "Terminal Protocols," IEEE Transactions in Communications, COM-28, pp. 585-593, April 1980.

15. Tanenbaum, Andrew S., Computer Networks, Prentice-Hall, Inc., 1981

16. Magnee, F. "A Survey of Terminal Protocols", Computer Networks, Vol. 3, pp. 299-314, November, 1979.

17. Stack, Thomas R. and Dillencourt, Kathleen A. "Protocols for Local Area Networks", IEEE 1980 Trends and Applications: Computer Network Protocols, pp. 83-93, May 1980.

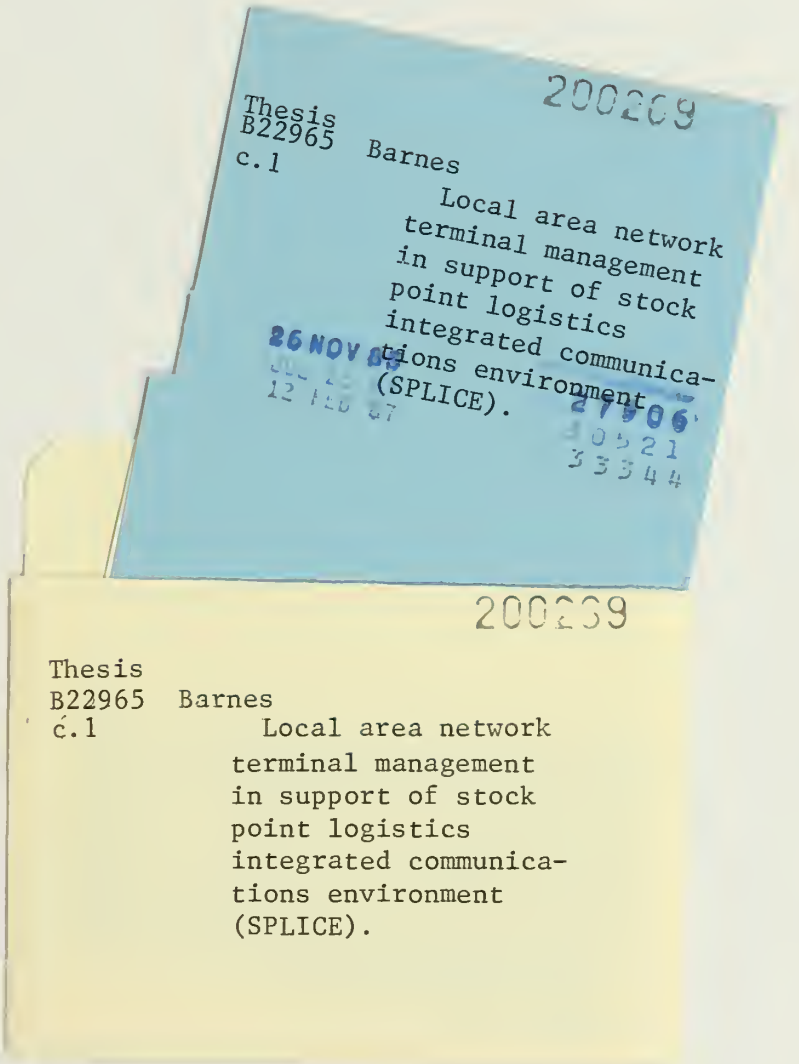
18. Schindler, Sigram; Flasche, Ute; and Bormann, Carsten, "OSI - Level 6 Protocols", Computer Communications, Vol. 5, No. 2, pp. 79-95, April, 1982.

19. Hillsberg, B.L. "Generic Terminal Support", Association for Computing Machinery, Operating Systems Review, Vol. 15, No. 2, April 1981.

20. Gerritsen, Rob "A Preliminary System for the Design of DBTG Data Structures", Communications of the ACM, Vol. 18, No. 10, pp. 551-556, October 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Professor N.F. Schneidewind, Code 54SS Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	2
4.	Professor Norman R. Lyons, Code 54LB Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
5.	Naval Postgraduate School Computer Technology Curricular Office Code 37 Monterey, California 93940	1
6.	LCDR J. D. Barnes, SC, USN Commander, Naval Supply Systems Command Code 0415A4 Washington D.C. 20376	2
7.	Ms. Mary Willoughby P.O. Box 94 Mendocino, CA 95460	1
8.	LCDR Dana Fuller Commander, Naval Supply Systems Command Code 04151 Washington D.C. 20376	1
9.	LCDR Ted Case Fleet Material Support Office Code 94L Mechanicsburg, PA 17055	1


200209

Thesis
B22965 Barnes
c.1

Local area network terminal management in support of stock point logistics integrated communications environment (SPLICE).

26 NOV 88
12 FEB 87
27506
10521
33344

200209

Thesis
B22965 Barnes
c.1

Local area network terminal management in support of stock point logistics integrated communications environment (SPLICE).

thesB22965
Local area network terminal management i
3 2768 002 01430 0
DUDLEY KNOX LIBRARY